Python - For Loops

Sandra Kuebler

Dept. of Linguistics, Indiana

Fall 2018

For Loops

Iteration

For loops allow us to iterate over each element of a set or sequence

Syntax:

```
for <var> in <set>:
    do ...
    do ...
```

Example

```
words = ['a', 'rose', 'is', 'a', 'rose', 'is',
                    'a', 'rose']
for w in words:
    print w
```

The Function Range

Iteration

Python has a built-in function, range, which allows us to generate a list containing the numbers specified in the range.

```
range(0, 10)
```

Range in Loops

```
for number in range(0, 10):
    print number
```

Range and Dictionaries


```
dict = {'NN' : 5, 'PRP' : 13, 'VBZ' : 4}
for key in dict:
    print key, 'occurred', dict[key], 'times.'
```

```
dict = {'NN' : 5, 'PRP' : 13, 'VBZ' : 4}
for key, val in dict.items():
    print key, 'occurred', val , 'times.'
```

Iteration

Caution: The loop will make sure that you will look at every key, but not in which order.

Breaking out of Loops

- `break`: stops the execution of the loop, independent of the test

```
for x in range(0,1000):
    if (x % 7) == 0:
        print 'first number divisible by 7:', x
        break
```

- `continue`: skip the rest of the loop body, but continue with the loop

```
for x in range(0,1000):
    if (x % 7) == 0:
        print x
    else:
        continue
```

Loop Else

Iteration

else can follow a loop, then it is executed in case the loop is NOT exited by break.

```
for x in range(0,1000):
    if (x % 7) == 0:
        print x
        break
else:
    print 'not found'
```